Analysis and Experimental Verification of Frequency-Based Interference Avoidance Mechanisms in IEEE 802.15.4

Lieven Tytgat, Opher Yaron, Sofie Pollin, Ingrid Moerman, and Piet Demeester, *Fellow, IEEE*

***Abstract*—More and more wireless networks are deployed with overlapping coverage. Especially in the unlicensed bands, we see an increasing density of heterogeneous solutions, with very diverse technologies and application requirements. As a consequence, interference from heterogeneous sources—also called cross-technology interference—is a major problem causing an increase of packet error rate (PER) and decrease of quality of service (QoS), possibly leading to application failure. This issue is apparent, for example, when an IEEE 802.15.4 wireless sensor network coexists with an IEEE 802.11 wireless LAN, which is the focus of this work. One way to alleviate cross-technology interference is to avoid it in the frequency domain by selecting different channels. Different multichannel protocols suitable for frequency-domain interference avoidance have already been proposed in the literature. However, most of these protocols have only been investigated from the perspective of intratechnology interference. Within this work, we create an objective comparison of different candidate channel selection mechanisms based on a new multichannel protocol taxonomy using measurements in a real-life testbed. We assess different metrics for the most suitable mechanism using the same set of measurements as in the comparison study. Finally, we verify the operation of the best channel selection metric in a proof-of-concept implementation running on the testbed.**

***Index Terms*—IEEE 802.11, IEEE 802.15.4, interference avoidance, MAC, medium access control, RDT, receiver directed transmission, WiFi, wireless sensor networks, ZigBee.**

I. Cross-Technology Interference Avoidance: Why and How?

IT IS increasingly hard to imagine a world without wireless communication. Today, we experience an exciting time given the emergence of the Internet of Things, which will allow any identifiable object in the world to communicate. Most objects will connect wirelessly, for obvious reasons. Hence, we can safely assume that the number of wireless devices will continue to grow exponentially [1]. Not only does the quantity of devices grow, but also the application domains diversify. Different application domains impose different requirements on the network, e.g., the quality of service (QoS) it needs to deliver, or the limitation on power consumption of network nodes that operate on batteries. These diversifying requirements can no longer be supported by a single wireless technology. Even more, within a single environment, multiple wireless technologies are being deployed in order to fulfill the applications needs. Hence, coexistence of different technologies is becoming increasingly important.

The coexistence of different technologies is particularly challenging when they share the same frequency band. Representative of such a situation are the unlicensed frequency bands, which are used by an increasing number of wireless technologies. As a result, different technologies that have not been designed to coexist need to operate in the same frequency bands, leading to reduced reliability of these technologies. A typical example, on which we focus in this paper, is the coexistence of IEEE 802.11 (WiFi) and IEEE 802.15.4 (ZigBee) networks. These technologies have very diverse application domains, but are typically deployed in identical surroundings such as homes, offices, and public buildings. It is shown in numerous studies that ZigBee suffers significant increase in packet loss rates in the presence of WiFi interference [2]–[5].

Cross-technology interference avoidance aims to avoid this interference in three domains—time, frequency, and space. Space-based frequency avoidance is not an option, for we need all sensor nodes to operate at the location they are in, and we do not want to lower WiFi transmit power since this results in decreased WiFi performance. Time-based interference avoidance between WiFi and ZigBee has already been studied. In [5], they experimentally prove that WiFi does not back off at all for IEEE 802.15.4, even for very strong ZigBee signal strengths. However, in [6], they state that WiFi can back off within a certain range, although it still creates collisions due to the slow clear channel assessment (CCA) of IEEE 802.15.4. Indeed, the WiFi standard [11] states that WiFi can implement preamble-based CCA resulting in increased intratechnology detection sensitivity but removing cross-technology detection capabilities altogether, or energy-based CCA that has lower intratechnology detection sensitivity but can also detect other technologies under some scenarios. Hence, depending on the implementation, WiFi might or might not be able to back off for

Manuscript received August 10, 2012; revised January 23, 2013 and July 01, 2013; accepted December 16, 2013; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Puthenpura. This work was supported by the European Union's Seventh Framework Programme FP7/2007-2013 under Grant Agreements No. 257542 (CONSERN project) and No. 258301 (CREW project), the IWT under projects ESSENCES and SYMBIONETS, iMinds under the project NGWINETS, and Research Foundation Flanders (FWO).

L. Tytgat, O. Yaron, I. Moerman, and P. Demeester are with iMinds, IBCN, Ghent University, 9050 Ghent, Belgium (e-mail: lieven.tytgat@intec.ugent.be; opher.yaron@intec.ugent.be; ingrid.moerman@intec.ugent.be; piet.demeester@intec.ugent.be).

S. Pollin is with the ESAT-TELEMIC, KU Leuven, 3001 Leuven, Belgium (e-mail: sofie.pollin@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2014.2300114

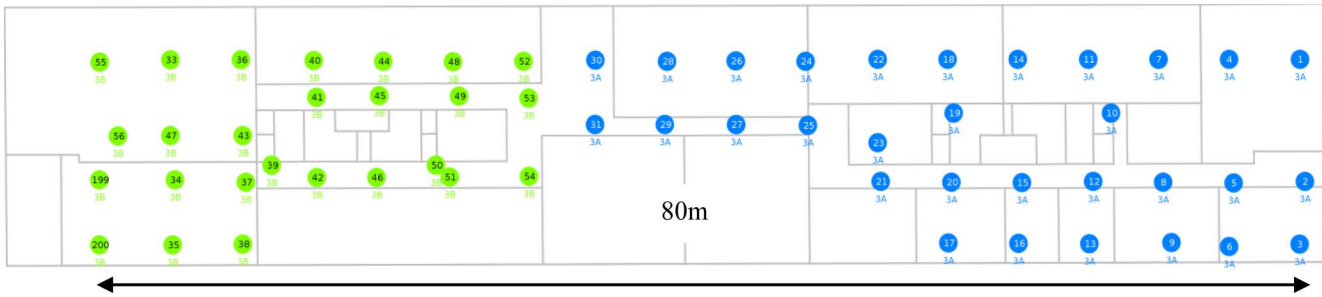

Fig. 1. Third floor of the iMinds w-iLab.t wireless testbed.

IEEE 802.15.4 within a certain range. In [7], we have solved this issue by adjusting the WiFi CCA, making it sensitive for WiFi as well as IEEE 802.15.4. As IEEE 802.15.4 networks cannot always rely on advanced WiFi CCA capabilities, there is still a need for coexistence solutions that do not rely on such enhanced WiFi capabilities. In [8], the authors present a method to exploit the typical bursty behavior of WiFi and reduce the amount of transmissions during a WiFi traffic burst. However, in case of continuous high-throughput WiFi networks, the throughput drops drastically. In such a scenario, it is simply favorable to avoid the occupied frequencies altogether. Hence, in this paper we study interference avoidance in the frequency domain, i.e., mechanisms that attempt to direct concurrent transmissions in co-located networks to different frequencies.

More specifically, we focus on multichannel protocols—in which individual nodes of a single network may operate on different channels. A plethora of multichannel protocols exists in the literature. Multichannel protocols are typically used to increase throughput by exploiting frequency-based parallelism. Within a cross-technology interference avoidance context, the maximum goodput (throughput times packet success rate) per channel is lowered due to the packet loss incurred by cross-technology interference. Typical sensor network applications require a low throughput and a long battery lifetime. Therefore, within sensor networks, the focus is usually on reliability and not on throughput. Hence, we focus on minimizing the amount of packet loss due to the interference received from other technologies. However, the relative advantages and disadvantages of multichannel protocols with respect to packet loss rates due to cross-technology interference have not been studied so far.

Therefore, in Section II, we analyze the wireless environment of a typical wireless sensor network, discuss related work, propose taxonomy for multichannel protocols, and compare different channel selection mechanisms defined in the taxonomy using testbed-based benchmark experiments. These experiments identify the Receiver Directed Transmission (RDT) protocol [17] as having superior properties. Although RDT is the most promising protocol, it lacks a channel selection metric. Hence, in Section III, we evaluate the performance of common channel selection metrics when applied to RDT using the same testbed-based benchmark experiments as in Section II and show there is opportunity for improvement. For that reason, we propose a new channel selection metric specific for RDT and verify its operation, again based on the same benchmark experiments. In Section IV, we elaborate on the proof-of-concept implementation and verify its runtime implementation on the testbed. Section V looks at future research, while we conclude this paper in Section VI.

II. Frequency-Based Interference Avoidance

A typical WiFi–ZigBee coexistence environment is an office building. ZigBee devices can be used for monitoring and control functions such as access control, HVAC monitoring and control, fire detection, etc., while WiFi is used for wireless Internet connectivity. A typical ZigBee network therefore needs to maintain the needed QoS within such an environment.

A. Home/Office Wireless Environment Characteristics

A thorough analysis of the time/space/frequency characteristics of the interference in a typical ZigBee environment aids in selecting the protocol that minimizes packet error rate (PER) in the ZigBee network. We measured the interference on the third floor of the iMinds w-iLab.t testbed [26] using the ZigBee nodes. This testbed is located in a 20×80-m^2 office building and consists of 200 nodes spread across three floors. Its third floor is depicted in Fig. 1.

Figs. 2 and 3 show interference measurements across the length of the building for all ZigBee channels during nighttime and daytime, respectively. Fig. 2 confirms that interference is local by nature. Moreover, there is at least one channel available with low interference levels across the building, for example channel 26. A single channel can therefore be selected that will result in relatively low perceived interference. However, Fig. 3 shows that during daytime, there is no single channel that has low interference throughout the building. Hence, we conclude that the interference environment is highly dynamic.

B. Multichannel Protocol Taxonomy

A multichannel protocol must guarantee that transmitter and receiver are on the same channel at the same time so that communication can take place. Every multichannel protocol is hence composed of three major components: 1) channel selection, which determines the channel at which to operate; 2) switching time scheduling, which determines when to actually switch to the selected channel; and 3) a mechanism to exchange/negotiate channel selection such as common control channel and distributed control channel, split-phase, etc.

Soua and Minet [31] propose a multichannel protocol taxonomy based on four questions: 1) What is the goal? 2) At what time is channel assignment done? 3) Which channel is selected? 4) How is channel assignment done? In [32], Incel proposes a taxonomy based on seven questions: 1) What is the

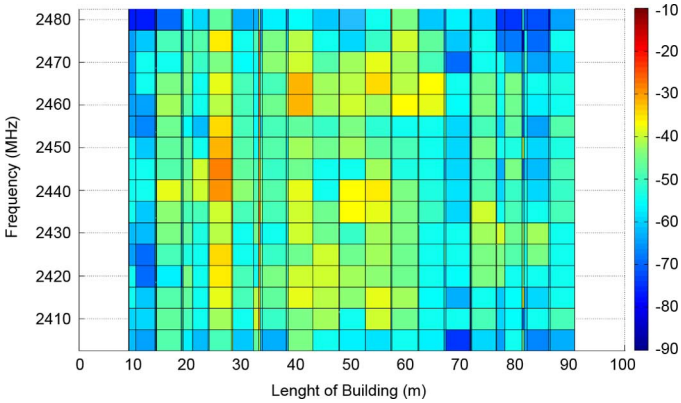

Fig. 2. Measured maximum interference levels—nighttime.

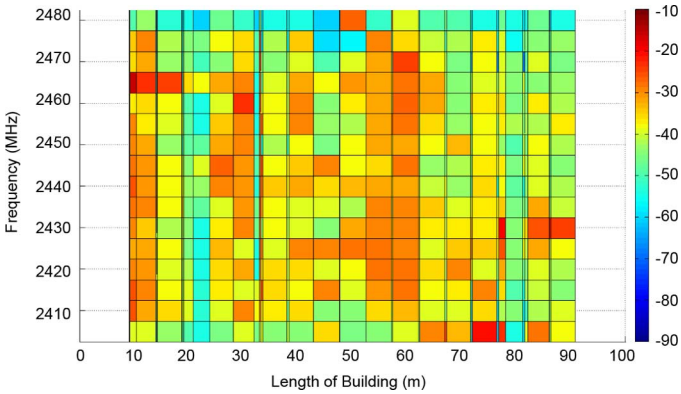

Fig. 3. Measured maximum interference levels—daytime.

channel assignment method? 2) Does the protocol need a control channel? 3) Is it a centralized or distributed protocol? 4) Do all nodes operate on one frequency at a given moment in time? 5) What is the type of medium access? 6) Does the protocol support broadcast? 7) What is the objective? Both works compare a number of protocols using their taxonomy. However, none of the studied multichannel protocols has cross-technology interference avoidance as goal. Even more, both taxonomies do not facilitate easy comparison of protocols within a cross-technology interference prone environment, nor do they allow prediction of protocol performance based on their classification.

Our taxonomy facilitates comparing the achievable performance under cross-technology interference by focusing only on the time and frequency behavior of protocols. In doing so, we do not incorporate the specific goal nor the mechanism to exchange/negotiate protocol information—also known as control traffic—into our taxonomy. However, our taxonomy aids in predicting the suitability of a given protocol type for a specific goal. Moreover, control and data traffic both have some time–frequency behavior, which might or might not be different. Our approach allows assessing the performance of control as well as data traffic, leading to a clear insight in the strengths and weaknesses of a complete protocol in heterogeneous interference scenarios. Fig. 4 shows our protocol taxonomy with the frequency behavior on the vertical axis (channel selection) and the time behavior on the horizontal axis (switching time).

Within our protocol taxonomy, we do not consider the used Medium Access Control (MAC) mechanism within each technology. MAC protocols typically intend to reduce intratechnology interference to an acceptable level. This might or might not result in reduced cross-technology interference. However, the multichannel protocol for optimal frequency-based cross-technology interference avoidance within a given environment can still be selected using the proposed multichannel protocol taxonomy, without loss of generality. Hence, a technology can still use its own medium access mechanism reducing the intratechnology collisions significantly, while the usage of a multichannel protocol reduces the cross-technology collisions.

We distinguish four different approaches to channel selection mechanisms: follow the master, (pseudo)random, internal metric-based, and external metric-based.

We define a node following the channel selection of another node—denoted the master—as a *follow the master* channel selection approach. In such a protocol, the master has some way of informing the slave of the channel selection to which it needs to adhere. A WiFi client is a typical example. It searches the channel of the access point (AP), connects to it, and remains on this channel. Another example is a Bluetooth slave device, which follows the hopping sequence of the master. It is informed of the hopping sequence it needs to follow by means of the master ID and a synchronization phase when joining the piconet [10]. A *pseudorandom* channel selection is not based on any ranking of channels and results in a flat distribution of the selection probability of any used channel. Hence, random, pseudorandom, round robin, etc., channel selections all fall into this category. A Bluetooth master is a typical example of a pseudorandom-hopping channel selection approach, while slave devices that are part of a piconet are obliged to follow the master's channel-hopping sequence. A metric-based protocol is defined as a protocol that creates some form of channel ranking and therefore can select a specific channel suited to support the goal of the protocol. We denote a channel metric as an *internal metric* when it is calculated without needing information from another node. A typical example of an internal metric is the channel selection of a WiFi AP. It selects its initial channel, based on some metric, independent of any client communication. In contrast, an *external metric* is a metric that can only be calculated through the usage of extra information from other nodes. Note that a distributed channel selection might use an internal (e.g., RDT) or external metric (e.g., Y-MAC), while a centralized channel selection by definition uses an external metric.

With regards to switching time, we also distinguish four different types: single shot, slotted, internal triggered, and external triggered. *Single shot* means that a node selects a channel at startup, and afterwards stays operating in that channel. A WiFi client that can only connect to one AP is a typical example. In contrast, a WiFi client that is able to connect to multiple APs on multiple frequencies may have a trigger causing it to switch to another AP, e.g., insufficient link quality from the current AP, an AP with higher received signal strength, etc. We call this approach *internal triggered* switching time. When the trigger is coming from another device, then we call it an *external trigger*. A typical *slotted* example is Bluetooth, wherein on every slot boundary, all nodes switch simultaneously to another channel.

At first glance, at least one type of multichannel protocol does not fit inside this taxonomy, namely Frequency-Division Duplex (FDD)-based protocols, of which a typical example is a regular cellular phone. In these protocols, the transmit frequency

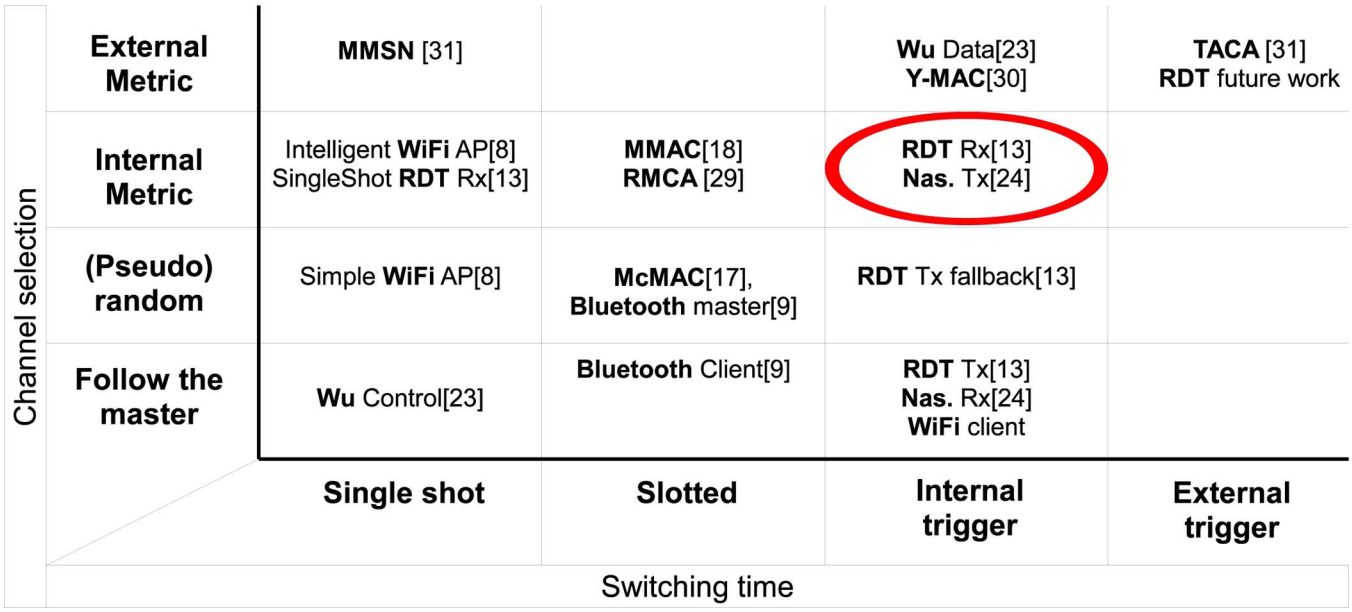

Fig. 4. Multichannel protocol taxonomy focusing on cross-technology interference avoidance capabilities with typical examples.

and the receive frequency are different, therefore seemingly not fitting the taxonomy. However, we simply separate the transmit and the receive channel selection, and both will again adhere to any of the time-frequency behaviors of our taxonomy. Hence, a cellular phone connected to one base station has a single-shot follow-the-master time-frequency behavior for the receiver as well as the transmitter, although they operate on different channels. Moreover, there are other protocols that use a different behavior for transmit and receive time-frequency behavior. For example, RDT [17] uses a triggered follow-the-master behavior for transmitting packets, and a triggered metric-based channel selection for receiving packets.

In [24], Nasipuri *et al.* propose a multichannel protocol that tries to minimize the collisions between WiFi nodes. This protocol determines the communication channel by assessing channel state before transmission. It remains on the current channel when it is free or hops to another channel when it is busy. The receiving nodes do not need to know the transmit channel, for they are continuously listening on all available channels. Hence, for the transmit side, this is an internal triggered switching time with an internal metric-based channel selection. For the receiver, this approach falls into the follow-the-master approach with an internal trigger since the receiver does not need any information from the transmitter. Wu *et al.* [23] propose to select the communication channel based on a usage list, which is updated through RTS/CTS like packets on a dedicated common control channel. Reliable communication on the control channel is guaranteed by employing two transceivers. One transceiver is dedicated to the control channel, while the other is solely used for data communication. Hence, the dedicated control channel is using a single-shot follow-the-master approach, while the data communication is using an internal triggered switching time with an external metric-based channel selection.

The operating principle of RDT is illustrated in Fig. 5. It separates the receive and transmit channels. Every node selects its own receive channel based on some metric. If it wants to transmit to another node, it does so on the receive channel of the destination. Hence, it switches its channel to the receive channel of the destination, transmits a packet, and returns to its own receive channel. In [17], RDT is proposed as a way to improve throughput. We, on the other hand, focus on its usage as an interference avoidance mechanism. In addition, we propose concrete mechanisms for selecting channels and for exchanging channel information between nodes that are not tackled in [17].

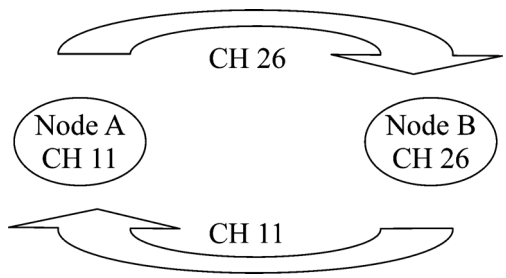

Fig. 5. Receiver Directed Transmission operating principle.

A number of protocols depicted in Fig. 4 are not yet discussed. However, discussing all of the available protocols is out of scope of this paper.

C. Taxonomy-Based Interference Avoidance Analysis

The protocol taxonomy together with the basic understanding of the interference environment allows us to compare and predict the interference avoidance performance of the different protocol classes.

We start of by determining the most suited channel selection mechanism. Out of the interference measurements, we conclude that there is no single channel available across the full length of the building. Hence, we can discard protocols that make all nodes operate on a single channel, which in our taxonomy fall in the follow-the-master class. During the daytime experiments,

we can clearly see that a large amount of channels receive a significant amount of interference. Therefore, a (pseudo)random approach, which essentially averages the packet loss incurred on each individual channel, will not perform as required. Protocols based on an external metric risk losing connectivity, as interference might become active, disconnecting one or more nodes from the network. In this case, it might not be possible to negotiate a new operating channel since the channel selection depends on communication with one or more other nodes. An effective interference avoidance protocol must allow the nodes to select channels in a distributed fashion, according to the local conditions without the requirement to exchange data with neighboring nodes. In our taxonomy, this is called internal metric-based channel selection.

We now focus on selecting the most promising switching time mechanism. Interference characteristics are dynamic over time. This conclusion is evident in the home/office environment, where people move around with their WiFi-enabled laptops and smartphones, and is also apparent from the comparison of daytime and nighttime measurements in Figs. 3 and 4. Due to the dynamism, we can predict that all single-shot-based protocols can result in a sudden drop in reliability. The single-shot class should hence be avoided. A slotted channel selection requires a node to select a new channel every predetermined interval. It needs to select a new channel even when the interference characteristics remain optimal on the current channel, resulting in a performance drop. A slotted switching time is therefore not desirable. An effective protocol must allow nodes to determine their own switching time according to changes in their own local environmental conditions, which in our taxonomy is referred to as triggered switching time. Moreover, nodes must be allowed to trigger a channel switch independently of other nodes and any ongoing communication with them. Hence, we predict that an internal trigger-based switching time will result in the most promising performance.

Hence, we conclude that an internal trigger-based switching time, combined with an internal metric-based channel selection, will most likely achieve best performance with regards to cross-technology interference avoidance. This conclusion is marked by a circle in Fig. 4. We now move on to identify the roles of different nodes. The signal-to-interference-plus-noise ratio (SINR) at the receiver determines the bit error rate of the transmission. The receiver should therefore be operating in the channel with the least interference. Hence, we forecast that RDT[17] will most likely be the best candidate for avoiding interference.

In Section II-D, we will experimentally compare the internal metric-based channel selection mechanism with (pseudo)random hopping and single-channel interference avoidance to verify the conclusions.

D. Experiment-Based Multichannel Mechanism Comparison

The taxonomy presented in Fig. 4 facilitates comparing the channel selection classes with respect to their ability to avoid interference. Within this section, we experimentally compare the performance of the different channel selection mechanisms on the iMinds w-iLab.t testbed using IEEE 802.15.4-based tmote sky sensor nodes [9]. This testbed is located in an office building where we cannot control the WiFi traffic of the regular

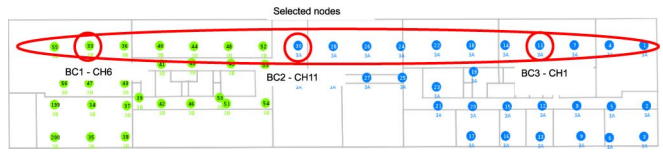

Fig. 6. RDT test setup with ZigBee nodes and WiFi interferers.

office users. However, during nighttime, the office is empty, and hence the level of background interference—which is primarily caused by beacons from idle WiFi APs—is relatively low.

For all tests, we selected a subset of nodes in one floor of the building that are aligned along the length of the building, as depicted in Fig. 6. This selection achieves a low average PER_Z between all nodes when there is no interference. We also selected three nodes to behave as WiFi interferers on different channels in order to emulate real-life WiFi network traffic. In all tests, all ZigBee nodes send an equal number of packets to all nodes.

Experiments were performed in three different interference scenarios, as follows.

BackGround interference (BG): In this scenario, experiments are performed at nighttime, and no extra interference is generated. Hence, only background interference created by the idle APs is present.

Emulated WiFi Interference (4.6 and 22.2 Mb/s): In this scenario experiments are also performed at nighttime, but extra controlled WiFi traffic is generated by the WiFi interferers in three different channels, as shown in Fig. 6. The three WiFi interferers are 802.11g devices that operate at a physical-layer speed of 54 Mb/s and a MAC payload of 1240 B. The different scenarios represent different requested packet rates: 4.6 Mb/s $=$ 471 packets/s $=$ 10% of maximum theoretical achievable throughput), 22.2 Mb/s $=$ 2220 packets/s $=$ 55% of the maximum theoretical achievable throughput. The transmit power of these devices is 10 dBm.

Real-Life Interference (Uncontrolled WiFi): In this scenario, experiments are performed at daytime during office hours. Real-life WiFi traffic is generated only by the regular office users and interferes with the ZigBee traffic of the experiment. Hence, we cannot control the loads on any of the WiFi devices.

In order to compare the performance of the different channel selection mechanisms, we create a benchmark of the environment, depicted in Fig. 7. Such a benchmark experiment is executed in all different interference scenarios. We collect link characteristics like PER, received signal strength, received interference, etc., between all nodes for all channels. This allows not only an easy comparison of the potential of the different channel selection mechanisms, but also the potential of specific metrics by emulating their operation *a posteriori*. The benefit of this approach is that different protocols and metrics can be analyzed based on an identical underlying set of measurements, facilitating comparability of the results. The downside is that we cannot compare triggered channel selections using this approach.

At the beginning of every experiment, all nodes tune to the first channel, channel 11, and measure the cross-technology interference—separating Signal and Interference in accordance to

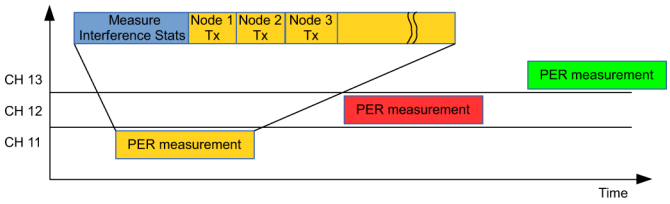

Fig. 7. Benchmark measurement sequence.

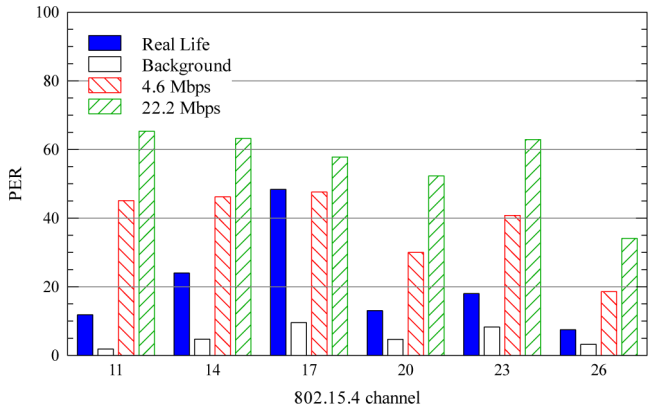

Fig. 8. Average PER_Z across all nodes for all channels and different interference scenarios. x-axis = ZigBee channel, y-axis = $\mathrm{PER}_Z(\%)$.

TABLE I
PER_Z COMPARISON BETWEEN INTERFERENCE AVOIDANCE MECHANISMS BASED ON THE BENCHMARK EXPERIMENTS. THE BEST IS HIGHLIGHTED

Protocol		Real Life	Background Interference	4.6 Mbps	22.2 Mbps
Best internal metric	Avg	6.87	0.46	16.22	24.37
	Worst	19.92	2.25	53.36	86.35
Best follow the master	Avg	7.43	1.04	18.18	33.91
	Worst	22.55	2.68	53.36	95.33
Random hopping	Avg	21.66	3.83	39.38	57.61
	Worst	30.52	7.21	71.02	94.81
Worst follow the master	Avg	54.52	7.16	47.47	67.20
	Worst	77.58	15.52	85.49	98.76
Worst internal metric	Avg	70.2	9.51	59.30	82.73
	Worst	80.6	15.52	92.95	98.76

Section III-C with a sample rate of 1/500 μs during 10 s. Statistics such as the minimum, average, and maximum interference+noise levels as well as a histogram of the measured power levels with 2 dB class width are collected. After the completion of this phase, each node broadcasts 1000 packets of 125 B at intervals of 12 ms, and all nodes report the packet error rate (PER_Z) for that sender. Once all nodes have completed their transmissions, they all switch to the next channel, and the same sequence is repeated. This is done for all ZigBee channels (11–26).

Fig. 8 shows the average PER_Z for a subset of channels in the different interference scenarios. It shows that real-life interference results in a high amount of packet loss. The background packet loss is significantly lower since the office space is abandoned. Table I is an aggregation of the measured statistics of the PER_Z between all pairs of nodes in all channels.

We compare three different channel selection mechanisms—namely follow the master (single channel), random (Bluetooth-like), and internal metric-based (RDT)—to a single-shot switching time. The internal metric-based approach allows every node to select its receive channel individually based on a metric.

The results are summarized in Table I, which shows the average PER across all nodes, as well as the average PER at the worst node. This worst node metric is important for the correct functioning of the full network. A single node with a high PER might not be able to deliver the needed QoS, resulting in application failure. A network that has a low average PER might therefore still be unable to support its application.

Table I shows that the lowest PER can be reached with an ideal internal metric-based approach when all channel information is known. The best follow-the-master selection is second best. In real-life background interference, 4.6 and 22.2 Mb/s scenarios, there is an average increase of respectively 8%, 126%, 12%, and 39% in comparison to best internal metric-based. Moreover, PER at the worst node is on average 29% higher than with the best internal metric. Random hopping is an approach that is not dependent on any channel selection metric since it hops in a random fashion across all used channels. As a result, it will average the PER of all used channels at every node, and hence never perform worse than the worst single channel, nor better than the best single channel. In contrast, a bad channel selection metric can potentially result in a worst-case channel selection that can happen with both other protocols. Pseudorandom hopping with blacklisted channels—e.g., Bluetooth 2.1—can reduce the packet loss in comparison to regular pseudorandom hopping. However, it will never improve upon the results of the best single channel as the whole network is always operating on one channel at any given moment. The worst follow-the-master solution obviously performs badly, followed by the worst internal metric-based channel selection that does it even worse. We can therefore conclude that a solution based on an internal metric is the most promising protocol, although the metric itself is crucial. For that reason, we will go in depth on the selection of an internal metric for RDT in Section III.

III. INTERFERENCE AVOIDANCE WITH RDT

A. RDT Runtime Metric Comparison

The metric that we want to optimize is the total average packet error rate in the ZigBee network. In RDT, every node selects its own receive channel and would ideally make this selection so as to minimize the average PER across all individual nodes. This minimum is reached when each individual node selects the channel with the least average PER. Determining the best channel could hence be achieved through measuring PER on all channels and selecting the best one. However, a reliable PER assessment requires a statistically relevant number of packets per pair of nodes on all channels, incurring a high amount of overhead traffic and no timely channel ranking. For practical implementation, a metric that can be measured instantaneously is preferred. Therefore, instead of measuring

TABLE II
PER_Z FOR COMMON CHANNEL SELECTION METRICS BASED ON THE BENCHMARK EXPERIMENTS

Metric		Real Life	Background Interference	4.6 Mbps	22.2 Mbps
Ideal PER	Avg	6.87	0.46	16.22	24.37
	Worst	19.92	2.25	53.36	86.35
min	Avg	26.85	6.30	39.40	56.79
	Worst	76.35	15.52	75.89	88.76
max	Avg	11.87	2.32	25.90	32.39
	Worst	22.58	9.51	72.28	95.33
avg	Avg	9.11	2.40	23.87	36.15
	Worst	28.26	9.51	72.28	95.33
Activity [23]	Avg	8.48	2.45	31.62	46.12
	Worst	24.94	7.20	72.28	95.33

PER, we try to build a channel ranking at runtime by measuring the interference levels on the different channels. Such a measurement, further referred to as a channel scan, samples the channel power for some time and calculates metrics from the collected samples.

A number of common metrics based on channel scans exist. We compare the performance of RDT when it uses these different channel selection metrics based on the link statistics and channel scan information collected during the same experiments that resulted into Table I. An overview of PER for all considered metrics in all scenarios is shown in Table II.

The "*ideal PER*" metric selects the channel with the least amount of PER, and thus results in the ideal channel selection. Hence, we will always compare the performance of a metric with this metric.

The "*min*" metric selects the channel where the minimal measured channel power is lowest as receive channel. Selecting the minimal measured channel power essentially results in measuring the radio's noise floor. We have selected the nodes in order to have sufficient link budget. In other words, the received signals are sufficiently above the noise floor of the radio, and hence this is not a good metric. In the real-life scenario, the resulting PER is increased by a factor 3.9 in comparison to the ideal PER metric.

The "*max*" metric selects the channel where the maximal measured channel power is lowest. This metric will avoid channels with high measured interference levels, independent of the load this interference level has. This leads to a good channel selection in case interference load is identical across all channels. Such an environment can be found in the background interference scenario, where it achieves identical performance as the PER metric. However, in the emulated interference and especially in the real-life interference, its performance drops drastically, where the average PER is increased with a factor 1.7 in comparison to the ideal PER metric.

The "*avg*" metric selects the channel with the lowest average measured channel power. Therefore, this will combine the effect of the interference power level and its load. As a result, we get fairly good performance under most circumstances. However, as can be seen in the worst node comparisons, some nodes select a less-than-optimal channel, which can be improved. The real-life PER is a factor 1.3 higher in comparison to the ideal PER metric.

The "*activity*" metric is a metric proposed in [23]. They propose to use the following metric and select the channel with the lowest "activity":

$$\text{Activity} = 100 \times \frac{\text{avg} - \min}{\max - \min} \tag{1}$$

with $\min$, avg, and $\max$ the minimum, average, and maximum measured channel power level.

This metric achieves good performance under most scenarios. It improves upon the avg metric with 7% in the real-life scenario. However, the PER achieved is still a factor 1.23 higher than with the ideal PER metric.

Out of this comparison, we conclude that the Activity metric is the best metric up to now. However, this metric results in a 1.23-times higher average PER in the real-life case than the ideal PER metric. Therefore, we create a new metric that comes closer to the performance of the ideal PER metric.

B. Building a New RDT Metric

We assume that the link budget of all transmitters is sufficiently high to guarantee negligible packet loss if no interferer is active. Moreover, for the sake of simplicity, we assume that no packet errors are caused by collisions between Zigbee packets.

In this paper, we focus on the interference of WiFi to ZigBee. The CCA of WiFi, when configured to energy-based CCA, may cause WiFi to backoff for ZigBee under specific scenarios. However, typical WiFi cards do not backoff for ZigBee at all because they implement preamble-based CCA [5]. Hence, the stochastic arrival processes of WiFi packets from all WiFi interferers are independent of any Zigbee activity, and we assume them to be identically distributed.

Fig. 9(a) shows that a packet that does not collide with interference is received with a sufficiently high signal-to-noise ratio (SNR), resulting in a negligible PER. In (b), the packet is interfered by interference level I1. The BER across the full packet in this case depends on the signal-to-interference ratio (SIR) between received signal strength 2 (S2) and Interference signal strength 1 (I1). I1 is received at low energy, resulting in a sufficiently high SIR, which we assume allows this packet to be received correctly with high probability. Case (c) depicts a collision between S2 and the stronger received interference I2, resulting in a low SIR and hence a low probability of successfully receiving the packet. In case (d), the signal has level S1, which is sufficiently above I2 to be successfully received with high probability. We conclude that when a specific packet is interfered, its successful reception depends on the signal levels of the transmitter and the interference at the receiver.

The PER as result of a specific SIR equals the expected packet error rate given a collision with this SIR multiplied by the probability of this SIR occurring. The total expected PER of a single receiver–transmitter pair $(\mathrm{E}\left(\mathrm{PER}(R,T)\right))$ can now be written as

$$\mathrm{E}\left(\mathrm{PER}(R,T)\right) = \int \mathrm{PER}(s)\,\Pr(s)ds \tag{2}$$

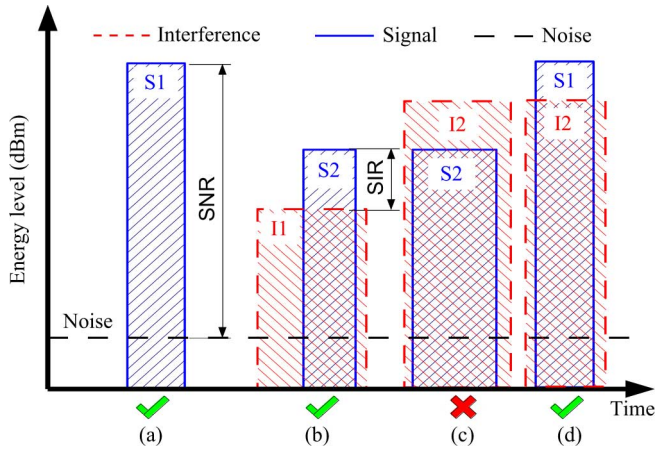

Fig. 9. Different interference scenarios: (a) strong S1 not interfered; (b) weak S2 interfered by weaker interference I1 does not result in packet loss; (c) S2 interfered by stronger I2 results in packet loss; (d) S1 interfered by I2 does not result in packet loss.

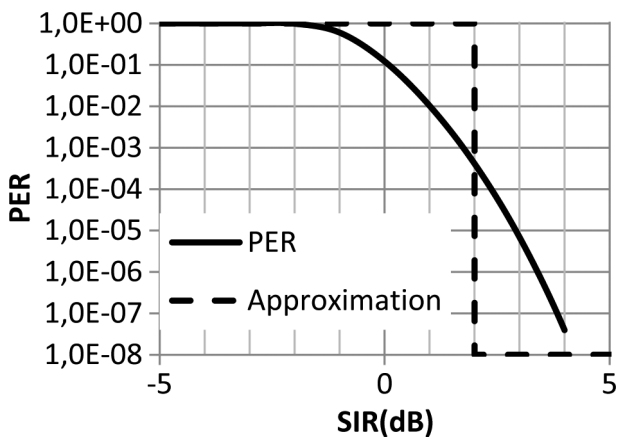

Fig. 10. PER for 100-B ZigBee packets.

with s the received SIR, and $\Pr(s)$ the probability distribution of SIR. Out of [10], we calculate the ZigBee PER versus SINR, depicted in Fig. 10.

Fig. 10 shows that the difference in SINR between 0.01% packet loss and 10% packet loss is 3 dB. In order to simplify our model, we neglect this 3 dB and approximate PER_Z as a step function dependent on the SIR

$$\mathrm{PER}(s) = H(s - Th_{\mathrm{SIR}}) \tag{3}$$

with $H(x)$ the Heaviside step function and Th_{SIR} the SIR threshold for good reception that we set at 2 dB since this results in less than 0.1% packet loss. Formula (2) can now be written as

$$\mathrm{E}\left(\mathrm{PER}(R,T)\right) \approx \int_{-\infty}^{Th_{\mathrm{SIR}}} \Pr(s)ds. \tag{4}$$

Below the threshold that we set at 2 dB, $E(\mathrm{err}|\mathrm{coll}) = 1$, above the threshold the $E(\mathrm{err}|\mathrm{coll}) = 0$. Note that in [13], Maheshwari *et al.* show that in an intratechnology interference context the usage of a full PER calculation is more accurate than a PER approximated by a threshold. The interference in both [10] (theoretical model) and [13] (empirical model) is fully

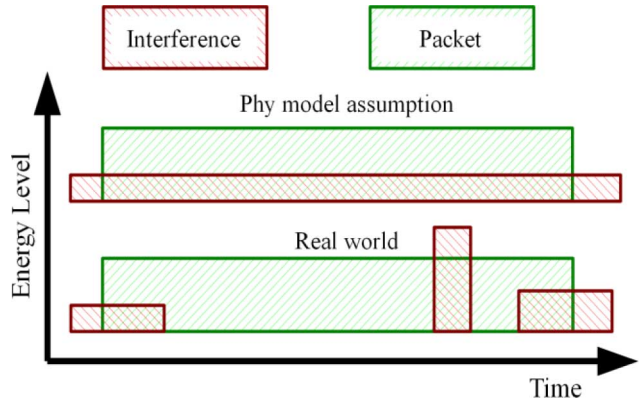

Fig. 11. Physical model assumption versus real-life cross-technology interference.

overlapping with the packets. However, in a cross-technology case, this assumption is not necessarily valid since the interference might only partially overlap with a packet, as depicted in Fig. 11. Hence, we cannot conclude to which extent the thresholding approximation impacts the accuracy of PER in the cross-technology case. Although the threshold-based approximation may be less accurate, it is the preferred model in view of implementation complexity since it allows a simple binary decision.

SIR equals to Signal strength minus Interference strength in logarithmic scale, leading to

$$\begin{aligned}\mathrm{E}\left(\mathrm{PER}(R,T)\right) &\approx \Pr(S - I \leq Th_{\mathrm{SIR}}) \\ &\approx \Pr(I \geq S - Th_{\mathrm{SIR}}).\end{aligned} \tag{5}$$

Formula (5) depicts that the packet loss on a link can be estimated by assessing the probability that the interference power at the time of packet transmission is higher than the signal power minus a threshold. We want to stress that in this formula, S is relatively static, while I is very dynamic. Therefore, the time behavior of the sum of all interferences determines the estimated PER.

We can measure the received signal strength for each transmitter. Hence, creating a histogram of the interference power levels allows us to assess this probability, and thus estimate PER. Fig. 12 depicts the histogram of the measured power levels on channels 14 and 26. From this histogram, we can easily estimate PER for any values of R and T.

The best receive channel is the channel where the average weighted expected PER of all neighboring nodes is lowest

$$\mathrm{E}\left(\mathrm{PER}(R)\right) \approx \frac{\sum_{i=1}^{N} \alpha_i \times \mathrm{E}\left(\mathrm{PER}(R,T)\right)}{\sum_{i=1}^{N} \alpha_i} \tag{6}$$

with α_i the weight on the estimated PER of a specific transmitter. Within the experiments, we assume the weight of all transmitters to be identical. We denote $\mathrm{E}(\mathrm{PER}(R))$ of (6) as the Received Signal to Interference Strength-based Thresholding *(ReSIST)* metric.

During startup, a node does not know the received signal strength of its neighboring nodes. As a consequence, we cannot rely on the ReSIST metric since no received signal strengths are known. Therefore, we bootstrap the channel selection by assuming a fixed received signal level from all nodes. A signal

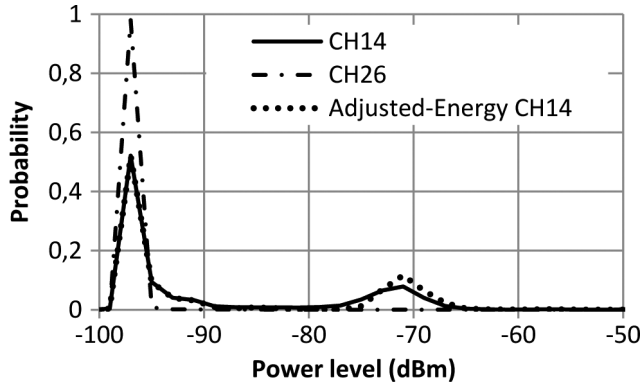

Fig. 12. Probability density function (histogram) of the measured interference power for different ZigBee channels.

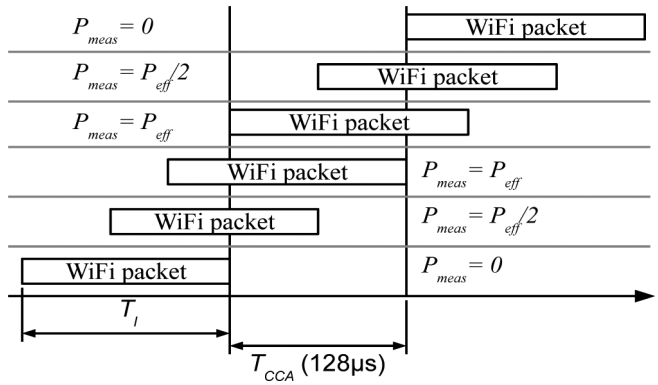

Fig. 13. Measured versus effective in band power.

TABLE III
DEFAULT PARAMETERS USED

	Zigbee	**WiFi**
MAC frame size (bytes)	100	1278
Datarate	250Kbps	54Mbps
Packet-time (μs)	127b: 4256 100b: 3392 50b: 1792 5b: 352	1Mbps: 10416 11Mbps: 1121 54Mbps: 212
T_{CCA}(μs)	128	4

strength of 10 dB above the receivers noise floor can be reached by every node within about 1/3 of the maximum communication range. Hence, we set the threshold at 10 dB above noise floor as it allows a normal operation of the network in most circumstances. Within the remainder of this paper, we refer to this metric as the Fixed Threshold *(FiT)* metric.

C. IEEE 802.15.4 Transceiver-Based Interference Assessment

In Section III-B, we elaborated on the theory how to determine the best channel through interference power measurements. In real life, the channel power measurements are not perfect. More specifically: 1) the power measurements include interference as well as signal and noise, while these should be separated in order to assess the resulting PER; and 2) the channel sample times are not necessarily small compared to the WiFi packet length (T_{CCA} in Table III). We will now determine the effects of, and solutions to, these nonideal measurements.

1) A regular Zigbee radio can return the power measured in the current channel in accordance with the IEEE 802.15.4 standard [10]. This measured power equals the sum of $\mathrm{Signal} + \mathrm{Interference} + \mathrm{Noise}$. Within this work, we neglect noise since we assume it does not result in packet loss. However, we still need to separate Signal from Interference. Two approaches can be identified: a) We can make certain that no signal is present during the power measurement. However, this implies not only that the network cannot operate during channel assessment times, but also that all 802.15.4 devices are under our control. b) We can separate signal samples from interference samples during execution of a channel scan. This can be achieved by using the preamble detection functionality of the radio. More specifically, the CC2420 radio used on the Tmote Sky can be configured to perform CCA based on either measured power level, or ZigBee preamble detection. Before starting a channel scan, we configure the CCA mode to ZigBee preamble detection. Before and after each power measurement, we check if the radio assesses the channel as busy or not and drop the measurement if any of the checks is positive. The remaining samples will predominantly contain only interference and noise.
2) The channel sample time of ZigBee equals 128 μs (to be denoted T_{CCA}), and the measured power is averaged across this window. A WiFi packet lasts between 28 μs and 12.4 ms. However, for the sake of simplicity, we initially assume all WiFi packets last at least 128 μs. The measured power in a sample will deviate from the effective interference power in case an interference signal starts or ends during the measurement window, as depicted in Fig. 13. Assuming the start and end of the interference is independent with respect to the start and end of the measurement window results in a uniform distribution of the overlap between measurement window and WiFi interference.

The total timeframe where WiFi packet energy is measured equals $T_{\mathrm{I}} + T_{\mathrm{CCA}}$. The sample will result in the effective signal power only when the CCA window fully overlaps with the WiFi packet, therefore removing $2 * T_{\mathrm{CCA}}$ from the total timeframe. The following equation calculates the probability of a sample returning the effective interference power for a fixed interference length:

$$\Pr(P_{\mathrm{meas}} = P_{\mathrm{eff}}) = \frac{T_{\mathrm{I}} - T_{\mathrm{CCA}}}{T_{\mathrm{I}} + T_{\mathrm{CCA}}}. \tag{7}$$

with P_{meas} the measured interference power, P_{eff} the real interference power, T_{I} the interference packet length, and T_{CCA} the measurement time.

Hence, the remaining part of the measurements $(1 - \Pr(P_{\mathrm{meas}} = P_{\mathrm{eff}}))$ will result in lower measured interference power. Fig. 14 depicts the resulting deviation of the measured

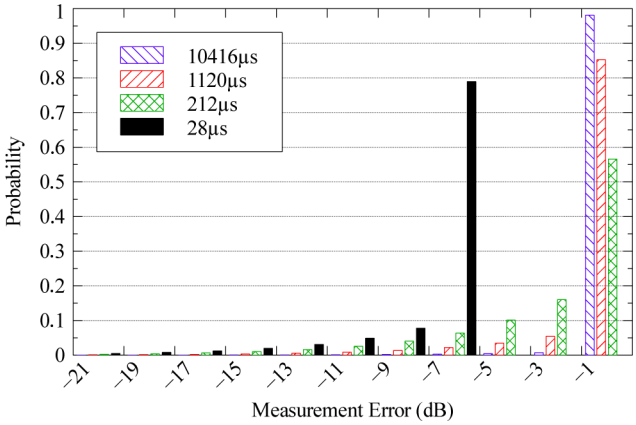

Fig. 14. Measurement error due to the long measurement window for different WiFi packet lengths for a class width of 2 dB.

power histogram with a class width of 2 dB for different interference packet lengths.

For example, Fig. 14 shows that 57% of all measurements of 212-μs-long interference packets will deviate less than 1 dB from the effective spectral power. Hence, 57% of the measurements will be captured inside the correct class. Furthermore, 16% will be between -1 and -3 dB or one class lower, 10% between -3 and -5 dB, etc. For the smallest (28 μs long) WiFi packets, no measurement will result in effective interference power since the interference is shorter than the measurement window. In fact, all measurement results are at least 6.6 dB $(= 10 * \log(28\ \mu s/128\ \mu s))$ lower than the effective power.

The measurement error for a specific interference packet length can now be compensated for to determine the actual interference power histogram. Starting from the highest class, the effective amount of samples that should have been inside this class can be calculated. For example, 57% of the effective samples for 212-μs interference lengths are actually measured in this class. Therefore, 1/0.57 times the number of samples measured in this class equals the effective number of samples, which an ideal measurement will measure. Now, the amount of samples that are measured in the lower classes—due to the long sample window—can be calculated and consequently removed from the respective lower classes. This calculation can be repeated recursively for all classes. We refer to the channel selection metric that is based on these adjusted energy measurements as Adjusted Energy ReSIST *(AE-ReSIST)*.

A plot of adjusted energy measurements of channel 14 is added in Fig. 12. It is clearly visible that the peak around -71 dB becomes higher, and the spill out in the lower classes is reduced, leading to a more accurate measurement. The downside is that we assume a specific fixed packet length, and hence introduce errors if this assumption is not correct. Moreover, the interference packet lengths will in general have a certain distribution that is not accounted for. However, future work could estimate this distribution by, e.g., machine learning techniques, exchange of WiFi packet length statistics between WiFi and ZigBee, etc.

D. Proposed Metric Comparison

In this section, we compare the performance of the three metrics proposed in Section III—namely FiT, ReSIST, and AE-ReSIST. The results of the experiments are shown in Table IV.

TABLE IV
PER_Z FOR NEWLY PROPOSED CHANNEL SELECTION METRICS BASED ON THE BENCHMARK EXPERIMENTS. THE BEST IS HIGHLIGHTED

Metric		Real Life	Back ground	4.6 Mbps	22.2 Mbps
Ideal PER	Avg	6.87	0.46	16.22	24.37
	Worst	19.92	2.25	53.36	86.35
FiT	Avg	7.74	2.44	23.53	35.82
	Worst	28.26	8.09	72.28	95.33
ReSIST	Avg	7.40	2.03	23.11	31.14
	worst	21.42	9.51	27.28	95.33
AE-ReSIST	Avg	7.40	2.07	23.15	31.27
	worst	21.42	9.51	72.28	97.46

FiT: Fixed Threshold-based interference classification (without using received signal strength information) selects the channel with the lowest FiT cost and improves upon all other metrics except in the background interference case. The real-life interference case results in an average PER a factor 1.13 higher than with the PER metric.

ReSIST: Received Signal and Interference Threshold-based interference classification (with received signal strength information) improves upon the performance of FiT in all scenarios. It results in a factor 1.07 higher PER than with the PER metric in the real-life scenario, which is small. However, the worst-case PER is usually different from the worst case of the PER metric. This is most likely due to other effects than WiFi interference significantly altering the effective link PER between nodes. More specifically, we believe this is due to multipath fading because we observed a high PER between a number of specific nodes (e.g., 1 and 4 in Fig. 1) in the background interference scenario that are physically only 5 m separated from one another. Out of the channel scans, we do not see significant WiFi interference strong enough to create this high level of PER. Therefore, multipath fading seems the most logical explanation, although true proof can only be found in a full electromagnetic analysis of the environment.

AE-ReSIST: Adjusted Energy ReSIST (ReSIST with adjusted energy measurements) performs identical to ReSIST in the real-life and background scenarios, but performs worse in the emulated scenarios, where ReSIST results in the best performance. The lack of improvement is due to the contradictory effect introduced by a model error and a measurement error. The model introduces an error by assuming that the Th_{SIR} is independent of the interferers on-time (i.e., The WiFi packet length). However, smaller interferer on-times result in a smaller average overlap between interference and packet (see Fig. 11), and thus a lower packet loss than predicted. In Section III-B, we show that the average measured signal level of the interference reduces with smaller interference on-times. Hence, the model overestimates the impact of smaller interference on-times, while the measurements, which serve as input to the model, underestimate the signal level of the interference for smaller packets, partially negating the overestimation the model makes. By reducing the measurement error, we remove the overestimation of the smaller interference on-times but keep the overestimation

the model makes, increasing the total error. Therefore, an improvement is to be expected only when correcting the measurement error as well as the error in the packet-loss model. However, building a precise cross-technology packet-loss model of which the parameters can be determined in a real-life scenario requires an in-depth study of the overlap between interference and the packet in a real-life environment, which is out of the scope of this paper.

IV. TinyOS-Based Implementation on Tmote Sky Hardware

A. Information Dissemination Mechanism

A packet can be received only if it is transmitted on the quiescent channel of its destination(s). The easiest way to achieve this is to transmit the packet on all channels. However, this multiplies the needed amount of transmissions and thus wastes battery power and creates additional interference. To avoid multichannel transmissions, it is necessary to inform the transmitter of the quiescent channel of the receiver.

We select two different mechanisms for distributing quiescent channel information to the surrounding nodes. The first mechanism is to periodically broadcast this information on all channels. This mechanism has the advantage of making sure that all nodes in the area are informed and also serves as a keep-alive packet with which the receiving nodes can update their neighbor database in case nodes lose connectivity. However, it is not efficient in terms of energy consumption, time incurred, and spectral usage. The second mechanism is to piggyback receive channel information on messages that are sent to neighboring nodes. This mechanism only costs a few additional bytes inside some of the transmitted packets. When a node decides to switch its receive channel while receiving a stream of packets, it can very quickly notify the sending node by piggybacking its acknowledgments. However, this mechanism cannot guarantee that all surrounding nodes know the quiescent channel.

The combination of both mechanisms overcomes both shortcomings. The periodic broadcasts make sure all surrounding nodes know the quiescent channel of the node. At the same time, piggybacking guarantees that nodes with which the transmitting node actively communicates are updated very quickly.

B. Implementation Architecture

The protocol is implemented on Tmote Sky nodes running TinyOS 2.1. It was implemented inside the radio driver as this makes it transparent to the higher layers. We have opted for a modular approach of three modules—namely RDT control, Channel Assessment, and a back-end database. The implementation independent settings—such as enable/disable RDT, allowable channels, channel scan time, etc.—can be governed by an external interface. The architecture of the implementation is shown in Fig. 15.

The RDT control module is responsible for the RDT information dissemination and the channel switching. The RDT control module piggybacks a packet with the receive channel given sufficient space is available in the packet. Periodic broadcasts are implemented by sending an empty packet with broadcast destination through the application-level active message interface.

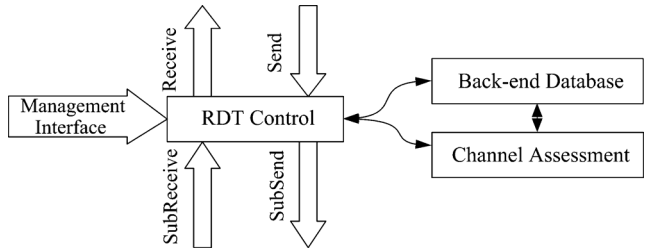

Fig. 15. RDT implementation architecture.

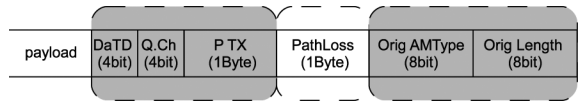

Fig. 16. Unicast piggyback trailer.

This packet is then automatically piggybacked since sufficient space is certainly available. The RDT control also switches the radio's channel when a packet needs to be transmitted.

The channel assessment module is responsible for selecting the receive channel of the node and returning the destination channel(s) of a packet. To resolve the receive channel, it performs the channel selection algorithm of Section III periodically. The channel switching module requests the destination channel(s) of a specific node to the back-end database module. If the receive channel is known, it is returned as a single destination channel. Otherwise, the packet needs to be transmitted on all channels that are in use by the system.

The back-end database module stores information regarding the surrounding nodes. Typical information includes receive channel, received signal strength, PER, time since last packet received/transmitted, etc. The receive channel information is used to supply the current receive channel of a node to the channel assessment module. The received signal strength is used to calculate the ReSIST metric. Although PER itself is not used in the protocol, it is used in the executed experiments for reporting purposes. The time since the last packet received or transmitted to a node is used to support mobility of the nodes and remove stale node data.

C. Packet Format Specification

Packets are piggybacked by adding extra trailers to the standard active messages created by TinyOS. When a packet is piggybacked, its AMType is overwritten with the RDT AMType of 255, thus allowing the receiver to distinguish between piggybacked packets and nonpiggybacked packets.

Two types of piggyback trailers are specified, one for unicast packets and the other for broadcast packets. The format of the unicast piggyback trailer is depicted in Fig. 16. The minimal trailer consists of the gray parts. These include the original AMType, the original packet length, the quiescent channel, and the transmit power of the packet. The DataType Definition (DaTD) field defines whether extra information is present in the trailer, e.g., measured path loss. Although the path loss is unused within this work, this can be used in future work in, e.g., transmit power adjustment.

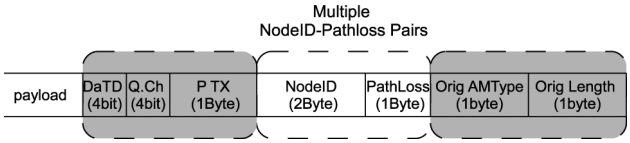

Fig. 17. Broadcast piggybacking trailer.

TABLE V
PER_Z OF SINGLE-SHOT AND TRIGGERED RESIST BASED ON RUNTIME IMPLEMENTATION

Metric	Real Life	Back ground	4.6 Mbps	22.2 Mbps
Single shot ReSIST	8.15	2.12	22.91	32.06
Triggered ReSIST	3.48	2.94	24.83	34.23

The format of the broadcast piggyback trailer is depicted in Fig. 17. It consists essentially of the same information included in the unicast trailer. However, since it reaches multiple destinations, specific information such as path-loss measurements for multiple nodes can be placed inside a single packet.

D. Implementation Results

In the online experiments, the RDT protocol implementation is used. Two different settings are used. The first is a single-shot channel selection setting. RDT scans all channels only at the beginning of the experiment and selects the best channel. No more channel switches are performed during the experiment. This setting allows for comparison between the RDT implementation and the RDT evaluated on the benchmark experiments. The second setting, triggered channel selection, allows RDT to dynamically switch channel selections during the experiment.

The experiments presented in Section III do not exploit the dynamism of RDT. The benchmarking experiment that is executed lasts nearly 2 h, resulting in a database that has average PER across a 2-h timeframe. Hence, we lose the time accuracy. The real-life implementation is set to scan the current channel every 15 s, allowing it to dynamically adjust to changing channel states. In the single-shot ReSIST scenario, the initial channel selection is maintained for the full experiment, while in the triggered scenario, RDT is allowed to change channels dynamically at runtime.

The results of the triggered ReSIST metric, which are shown in Table V, are significantly better in the real-life scenario in comparison to the single-shot results. The remaining scenarios are slightly worse than the benchmark-based scenarios. This can be explained by the dynamic nature of RDT in a static scenario. Deviations in the measurements might make the nodes hop to a channel with a higher PER for a short time, until it performs another channel scan that is worse than the best channel, and it hops back. However, in the real-life scenario, the channel states change significantly in comparison to the measurement deviations, resulting in channel hops to channels with better channel states.

V. FUTURE WORK

RDT is capable of coping with dynamic environments given it has relevant state information of all channels. However, a ZigBee node only operates on one channel at a given moment, and hence only the state information of the current channel is updated. This has as effect that the state information of the other channels becomes outdated. Updating these channel states can be done by temporarily switching the quiescent channel. However, this might result in a temporary deterioration of the PER. This tradeoff between exploration and exploitation, which can be solved optimally if the problem can be formulated as a multiarmed bandit problem, needs to be considered.

Sensor networks are usually battery-powered and therefore energy-sensitive. The current RDT implementation does not consider energy-saving mechanisms, commonly used in sensor networks. Hence, combining RDT with an energy-saving protocol is certainly an interesting topic. Moreover, RDT exchanges protocol information—which consumes energy, but also reduces the number of needed transmissions, which saves energy. The channel scan, combined with the path-loss information can also be used for transmit power adjustment. The channel scan information of the receiver can namely be used at the transmitter to determine the expected PER, resulting in minimal transmit power for a requested link PER. We have done an *a posteriori* comparison of different single-shot interference avoidance protocols as well as RDT metrics starting from identical benchmark experiments. Due to dynamism in the environment, a single-shot channel selection might not be maintainable across the full lifetime of a sensor network. However, comparing triggered channel selection protocols and metrics is extremely hard because multiple experiments, which are done at different times, are needed. Hence, extreme care needs to be taken that we compare the protocols and metrics and not the difference in the environment. Repeatability and reproducibility of wireless experiments is a hot topic that is addressed today by many researchers. We refer for instance to [37]. An in-depth comparison of triggered protocols and metrics thus remains an open issue.

VI. CONCLUSION

Coexistence of different wireless technologies is becoming an increasingly limiting factor in achieving the needed QoS with a certain technology. We show through measurements in an office environment that the interference created by WiFi on a ZigBee network is of a dynamic, local nature.

Using our proposed multichannel protocol taxonomy, we conclude that an internal metric-based channel selection combined with an internal trigger-based switching time is the most suitable packet-loss-reducing protocol in an office environment. We experimentally verify that an internal metric-based channel selection indeed performs best in real-life environments. It is able to reduce the average PER with a factor 3.43 and 1.73 compared to (pseudo)random channel selection and the best single channel, respectively. However, it can perform worse in case a wrong channel metric is used.

We therefore analyze the performance of commonly used metrics and show that a significant improvement is achievable. Hence, we propose a new metric—called ReSIST—and experimentally verify its operation. We show that our channel metric reduces the average PER with a factor 3.63, 1.60, 1.23,

and 1.14 in comparison to respectively min, avg, max, and activity [23] channel metrics in real-life cases. We also verify that our channel metric degrades with 7.7% compared to the situation where we have full channel information. Therefore, we proposed an improvement to ReSIST that reduces the measurement error incurred by IEEE 802.15.4-based channel assessments. However, we concluded that the performance did not improve as expected, as we reduce only one out of two contradictory errors, explained in depth in Section III-D. Finally, we verified our implementation of triggered ReSIST—which is able to switch channels dynamically at runtime—and conclude that in the real-life case, a PER reduction with a factor 2.34 in comparison to a single-shot channel selection is achievable.

REFERENCES

[1] M. Meekers, S. Devitt, and L. Wu, "Morgan Stanley Internet trends 04/12/2010 (Morgan Stanley research 2010)," Accessed Jan. 26, 2014 [Online]. Available: http://comunicaciondecrisis.wikispaces.com/file/view/Internet_Trends_041210+Morgan+Stanley.pdf/289548087/Internet_Trends_041210%20Morgan%20Stanley.pdf

[2] Z. Bin, L. Huan-Bang, H. Shinsuke, and K. Ryuji, "Clear channel assessment in integrated medical environments," *EURASIP J. Wireless Commun. Netw.*, vol. 2008, p. 821756, Jan. 2008, Art. no. 48.

[3] G. Thonet, P. Allard-Jacquin, and P. Colle, "ZigBee–WiFi coexistence white paper and test report," 2012 [Online]. Available: http://www.ZigBee.org

[4] Y. Wei, W. Xiangyu, and J.-P.M. G. Linnartz, "A coexistence model of IEEE 802.15.4 and IEEE 802.11b/g," in *Proc. 14th IEEE Symp. Commun. Veh. Technol. Benelux*, Nov. 2007, pp. 1–5.

[5] S. Pollin, I. Tan, B. Hodge, C. Chun, and A. Bahai, "Harmful coexistence between 802.15.4 and 802.11: A measurement-based study," in *Proc. 3rd CrownCom*, May 2008, pp. 1–6.

[6] C. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving WiFi interference in low power ZigBee networks," in *Proc. SenSys*, Zurich, Switzerland, Nov. 3–5, 2010, pp. 309–322.

[7] L. Tytgat, O. Yaron, S. Pollin, I. Moerman, and P. Demeester, "Avoiding collisions between IEEE 802.11 and IEEE 802.15.4 through coexistence aware clear channel assessment," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, p. 137, 2012.

[8] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: Exploiting WiFi white space for ZigBee performance assurance," in *Proc. 18th IEEE ICNP*, Washington, DC, USA, 2010, pp. 305–314.

[9] Moteiv Corporation, San Francisco, CA, USA, "TMOTE SKY datasheet," Accessed Aug. 06, 2012 [Online]. Available: http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf

[10] *IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std. 802.15.4, 2006.

[11] *IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11, 2012.

[12] *IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*, IEEE Std. 802.15.1, 2005.

[13] R. Maheshwari, S. Jain, and S. R. Das, "A measurement study of interference modeling and scheduling in low-power wireless networks," in *Proc. 6th ACMSenSys*, New York, NY, USA, 2008, pp. 141–154.

[14] L. Tytgat, M. Barrie, V. Gonçalves, O. Yaron, I. Moerman, P. Demeester, S. Pollin, P. Ballon, and S. Delaere, "Techno-economical viability of cognitive solutions for a factory scenario," in *Proc. IEEE DySPAN*, May 3–6, 2011, pp. 254–264.

[15] M. Barrie, L. Tytgat, V. Gonçalves, O. Yaron, I. Moerman, P. Demeester, S. Pollin, P. Ballon, and S. Delaere, "Techno-economic evaluation of cognitive radio in a factory scenario," in *Proc. PE-CRN*, May 9–13, 2011, pp. 52–61.

[16] A. W. Min, K. Kim, and K. Shin, "Robust cooperative sensing via state estimation in cognitive radio networks," in *Proc. IEEE DySPAN*, 2011, pp. 185–196.

[17] N. Shacham and P. King, "Architectures and performance of multichannel multihop packet radio networks," *IEEE J. Sel. Areas Commun.*, vol. SAC-5, no. 6, pp. 1013–1025, Jul. 1987.

[18] R. Balamuthi, H. Joshi, C. Nguyen, A. K. Sadek, S. J. Shellhammer, and C. Shen, "A TV white space spectrum sensing prototype," in *Proc. IEEE DySPAN*, 2011, pp. 297–307.

[19] P. V. Wesemael, S. Pollin, E. Lopez, and A. Dejonghe, "Performance evaluation of sensing solutions for LTE and DVB-T," in *Proc. IEEE DySPAN*, 2011, pp. 531–537.

[20] IMEC, Leuven, Belgium, "IMEC sensing engine development," 2012 [Online]. Available: http://www.imec.be/ScientificReport/SR2008/HTML/1225000.html

[21] Y. Xiao and J. Rosdahl, "Throughput and delay limits of IEEE 802.11," *IEEE Commun. Lett.*, vol. 6, no. 8, pp. 355–357, Aug. 2002.

[22] S. Shin, H. Park, S. Choi, and W. Kwon, "Packet error rate analysis of IEEE 802.15.4 under IEEE 802.11b interference," in *Proc. 3rd WWIC*, Xanthi, Greece, 2005, pp. 1186–1190.

[23] M. Hossian, A. Mahmood, and R. Jantti, "Channel ranking algorithms for cognitive coexistence of IEEE 802.15.4," in *Proc. 20th IEEE Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 13–16, 2009, pp. 112–116.

[24] H.-S. So, W. Walrand, and J. J. Mo, "McMAC: A parallel rendezvous multi-channel MAC protocol," in *Proc. IEEE WCNC*, Mar. 11–15, 2007, pp. 334–339.

[25] J. So and N. Vaidya, "Multi-channel MAC for Ad Hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proc. ACM Mobihoc*, May 2004, pp. 222–233.

[26] CREW Project, "w.iLab.t portal," Accessed May 15, 2012 [Online]. Available: http://www.crew-project.eu/wilabt

[27] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," in *Proc. 10th Int. Conf. Comput. Commun. Netw.*, 2001, pp. 432–439.

[28] S.-L. Wum, C.-Y. Linm, Y.-C. Tseng, and J.-P. Sheu, "A novel MAC protocol with on-demand channel assignment for multi-hop mobile ad-hoc networks," in *Proc. ISPAN*, 2000, p. 323.

[29] A. Nasipuri, J. Zhuang, and S. R. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *Proc. Wireless Commun. Netw. Conf.*, 1999, vol. 3, pp. 1402–1406.

[30] K. Bian, J.-M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 689–703, Apr. 2011.

[31] R. Soua and P. Minet, "A survey on multichannel assignment protocols in wireless sensor networks," in *Proc. IFIP Wireless Days*, Oct. 2011, pp. 1–3.

[32] O. D. Incel, "A survey on multi-channel communication in wireless sensor networks," *Comput. Netw.*, vol. 53, no. 13, pp. 3081–3099, Sep. 2011.

[33] Q. Yu, J. Chen, Y. Sun, Y. Fan, and W. Shen, "Regret matching based channel assignment for wireless sensor networks," in *Proc. IEEE ICC*, Cape Town, South Africa, 2010, pp. 1–5.

[34] Y. Kim, H. Shin, and H. Cha, "Y-MAC: An energy efficient multichannel MAC protocol for dense wireless sensor networks," in *Proc. IPSN*, St. Louis, MO, USA, 2008, pp. 53–63.

[35] Y. Wu, M. Keally, G. Zhou, and W. Mao, "Traffic-aware channel assignment in wireless sensor networks," in *Proc. 4th WASA*, 2009, pp. 479–488.

[36] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–13.

[37] CREW Project, "CREW Project homepage," Accessed Jun. 2013 [Online]. Available: http://www.crew-project.eu/,

Lieven Tytgat received the Master's degree in industrial science on electronics from the Hogeschool West-Vlaanderen (HoWest), Kortrijk, Belgium, in 2004, the Master's degree in industrial management from the Katholieke Universiteit Leuven (KULeuven), Leuven, Belgium, in 2005, and the Master of Science degree in electrotechnics from Ghent University (UGhent), Ghent, Belgium, in 2007, and is currently pursuing the Ph.D. degree in cross-network cooperation paradigms supporting co-located heterogeneous wireless networks at UGhent.

He joined the Ghent University research group on Internet Based Communication Networks and Services (IBCN, http://www.ibcn.intec.ugent.be) in 2007. As a member of this research group, he is also affiliated with the Interdisciplinary Institute for BroadBand Technology (iMinds, http://www.iMinds.be). He is involved in research projects focussing on wireless networks, software defined radio, and cognitive radio.

Opher Yaron received the Master's degree in mathematics from the Hebrew University, Jerusalem, Israel, in 1987, and the Doctor of Science degree in electrical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 1994.

He is the Chief Technology Officer of Track4c (http://www.track4c.com), a startup company that focuses on smart containers as an effective and affordable solution for transparent and secure supply chains. Prior to joining Track4c, he was a Senior Researcher with Ghent University, Ghent, Belgium, within the research group on Internet-Based Communication Networks (IBCN), which is also part of iMinds (formerly IBBT). He has more than 20 years of experience in the high-tech industry. He held various technical, managerial, and leadership positions, including at the vice-president level, for established and startup companies in the areas of data communications, Voice over IP (VoIP), advanced WLAN technologies, and software systems for mobile value added services. His research focused on performance and QoS in broadband communication networks; cognitive, self-organizing, and self-optimizing networks; and network architectures and protocols for heterogeneous mobile and wireless networks.

Sofie Pollin received the Ph.D. degree (with honors) on cross-layer energy and QoS optimization from Katholieke Universiteit Leuven (KU Leuven), Leuven, Belgium, in 2006.

For her Ph.D. studies, she cooperated with many researchers at imec, Leuven, Belgium, and worldwide, did an internship with National Semiconductor, Santa Clara, CA, USA, and was a Visiting Scholar with the University of California (UC), Berkeley, CA, USA. After the Ph.D., she continued her research on wireless communication, energy-efficient networks, cross-layer design, coexistence, and cognitive radio at UC Berkeley for 2 years. In 2008, she returned to imec to become a Principal Scientist in the green radio and cognitive radio teams. Since 2012, she has been a tenure track Assistant Professor with ESAT, KU Leuven, within the TELEMIC division. Her research centers around networked systems that create networks that are ever more dense, heterogeneous, battery-powered, and spectrum-constrained.

Dr. Pollin is a BAEF and Marie Curie Fellow.

Ingrid Moerman received the Electrical Engineering degree and the Ph.D. degree from Ghent University, Ghent, Belgium, in 1987 and 1992, respectively.

She has been a Professor with Ghent University. She is staff member of the research group on Internet Based Communication Networks and Services (IBCN, http://www.ibcn.intec.ugent.be), where she is leading the research on mobile and wireless communication networks. In 2006, she joined the Interdisciplinary institute for BroadBand Technology (iMinds, http://www.iminds.be), where she is coordinating several interdisciplinary research projects. At the European level, she is in particular very active in the FP7 FIRE (Future Internet Research and Experimentation) research area, where she is coordinating the CREW project and further participating in IP OpenLab, IP Fed4FIRE, IP OFELIA, STREP SPITFIRE, and STREP EVARILOS. In the FP7 research area on Future Networks, she is involved in IP LEXNET, STREP CONSERN, and STREP SEMAFOUR. She is author or coauthor of more than 500 publications in international journals or conference proceedings. Her main research interests include sensor networks, cooperative and cognitive networks, wireless access, self-organizing distributed networks, and experimentally supported research. She has long-standing experience in running national and EU research funded projects.

Prof. Moerman is an Associate Editor of the *EURASIP Journal on Wireless Communications and Networking* and Vice-President of the expert panel on Informatics and Knowledge Technology of the Research Foundation Flanders (FWO).

Piet Demeester (M'89–SM'98–F'09) received the Ph.D. degree in metal organic vapor phase epitaxy for photonic devices from Ghent University, Ghent, Belgium, in 1988.

He is a Professor with the faculty of Engineering, Ghent University. He is head of the research group "Internet Based Communication Networks and Services" (IBCN, http://www.ibcn.intec.ugent.be) that is part of the Department of Information Technology (INTEC), Ghent University, and that is also affiliated with the Interdisciplinary Institute for Broadband Technology (iMinds, http://www.iminds.be). After finishing the Ph.D., he established a research group in his area of study, working on different material systems (AlGaAs, InGaAsP, GaN). This research was successfully transferred to IMEC in 2002 and resulted in 12 Ph.D. degrees and 300 publications in international journals and conference proceedings. In 1992, he started research on communication networks and established the IBCN research group. The group is focusing on several advanced research topics: network modeling, design and evaluation; mobile and wireless networking; high performance multimedia processing; autonomic computing and networking; service engineering; content and search management, and data analysis and machine learning. The research of IBCN resulted in about 50 Ph.D. degrees, 1250 publications in international journals and conference proceedings, 30 international awards, and four spin-off companies.